

---

# **django-intranet Documentation**

*Release 0.2*

**lonyse**

**Nov 14, 2017**



---

## Contents

---

<b>1</b>	<b>Abstract</b>	<b>1</b>
<b>2</b>	<b>Table of contents</b>	<b>3</b>
2.1	Setup a new project . . . . .	3
2.2	Create a new module . . . . .	5
2.3	Utils . . . . .	10



# CHAPTER 1

---

## Abstract

---

`django-intranet` is the minimal codebase to create fast enterprise management applications. Based on Django-1.4 you start a project, add `intranet` and the other module you want to use. That's it, you are ready to go with your intranet.



## 2.1 Setup a new project

### 2.1.1 Install django-intranet

Create a project folder and enter it. Type these command lines.

```
$ sudo pip install django-intranet south
$ django-admin.py startproject myintranet
```

### 2.1.2 Configure the database

Configure the database and add ‘intranet’ and ‘south’ to your `INSTALLED_APPS` in `myintranet/myintranet/settings.py`.

Go to `myintranet/` and type:

```
$ python manage.py migrate
```

Now you can install one of the contrib module or your own.

### 2.1.3 Setup the URLs

```
# -*- coding: utf-8 -*-
from django.conf.urls import patterns, include, url
from django.core.urlresolvers import reverse_lazy
from django.conf import settings
from django.views.generic import RedirectView
from django.contrib import admin

from intranet.urls import urlpatterns as intranet_urlpatterns
```

```

admin.autodiscover()

# Default URLs patterns from intranet
urlpatterns = intranet_urlpatterns

# Add specific patterns
urlpatterns += patterns('',

    url(r'^$', RedirectView.as_view(url=reverse_lazy('dashboard'))),

    # Apps
    url(r'^prospect/', include('prospect.urls')),

    # Admin URLs
    url(r'^admin/', include(admin.site.urls)),
    url(r'^admin_tools/', include('admin_tools.urls')),

)

if settings.DEBUG:
    from django.conf.urls.static import static
    from django.contrib.staticfiles.urls import staticfiles_urlpatterns

    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
    urlpatterns += staticfiles_urlpatterns()

```

## 2.1.4 Configure the navbar

Create the file `templates/intranet/navbar.html`:

```

{% load intranet_extra i18n %}
{% load url from future %}

<div class="navbar navbar-fixed-top">
  <div class="navbar-inner">
    <div class="container-fluid">
      <a class="btn btn-navbar" data-toggle="collapse" data-target=".nav-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </a>
      <a class="brand" href="{% url 'dashboard' %}">{% intranet_title %}</a>
      <div class="nav-collapse">
        <ul class="nav">
          <li class="divider-vertical"></li>
          <li><a href="#"><i class="icon-user icon-white"></i>&nbsp;&nbsp;&nbsp;{% trans "Profile" %}</a></li>
          <li class="divider-vertical"></li>
          <li><a href="{% url 'prospect-list' %}">
            <i class="icon-road icon-white"></i>&nbsp;&nbsp;&nbsp;{% trans "Prospects" %}</a></li>
          <li class="divider-vertical"></li>
        </ul>
        {% if request.user.is_authenticated %}
        <p class="navbar-text pull-right"><a href="{% url 'logout' %}">{% trans %}

```



```

<ul class="nav pull-right"><li class="divider-vertical"></li></ul>
<p class="navbar-text pull-right">{% trans "Logged in as" %}&nbsp;<a href="#">
→{{ request.user.username }}</a></p>
    {% endif %}
</div><!--/.nav-collapse -->
</div>
</div>
</div>

```

## 2.2 Create a new module

Here are the few steps to create a *django-intranet* module.

### 2.2.1 Models

First of all create your models:

```

from django.db import models
from django.utils.translation import ugettext_lazy as _

class Category(models.Model):
    name = models.CharField(_('name'), max_length=50)

    def __unicode__(self):
        return u'%s' % self.name

    class Meta:
        verbose_name = _('category')
        ordering = ('name', )

class Prospect(models.Model):

    company_name = models.CharField(_('company name'), max_length=100)
    category = models.ForeignKey(Category, related_name="prospects")

    title = models.CharField(_('title'), max_length=5, blank=True)
    first_name = models.CharField(_('first name'), max_length=50, blank=True)
    last_name = models.CharField(_('last name'), max_length=50, blank=True)
    e_mail = models.EmailField(_('e-mail'), blank=True)
    phone_number = PhoneField(_('phone number'), blank=True)

    def __unicode__(self):
        return u'%s' % self.company_name

    class Meta:
        verbose_name = _('prospect')
        ordering = ['company_name']

```

### 2.2.2 URLs

You can automatically generate URLs and views for displays and treatments of your database: display of the list of your database table, display of the details of an entry, creation, modification and removal of an entry. To use this feature, you have to add the new generated URLs to your URL patterns:

```
from intranet.utils import get_default_model_url
from my_app.models import Category, Prospect

urlpatterns = get_default_model_url(Category)
urlpatterns += get_default_model_url(Prospect)
```

There are also a number of parameters you can customize:

- *form\_class*: a customized form view for your model
- *url\_name*: a string to name the URLs of your model in your application. By default it is the name of your model in lower-case.
- *prefix\_pattern*: a string or a regular expression to define a prefix before the URLs of your model. There is no prefix by default.
- *prefix\_name*: a string that will be used as a prefix before the name of your URLs (*url\_name*). There is no prefix by default.
- *list\_view*, *detail\_view*, *creation\_view*, *update\_view*, *delete\_view*: customized views for respectively the display of the list of your database entries, the display of one entry, the creation, update and removal of an entry.

For example, you could define the following URL pattern:

```
# Import your own vues.
from views import ProspectListView, ProspectDetailView

urlpatterns = get_default_model_url(
    Prospect,
    url_name = 'pro',
    prefix_pattern = '(?P<company_id>\d+)/',
    prefix_name = 'pre-',
    list_view = ProspectListView,
    detail_view = ProspectDetailView,
```

The previous code is equivalent to:

```
from views import ProspectListView, ProspectDetailView
from django.conf.urls import patterns, url

urlpatterns = patterns(url(
    r'^(?P<company_id>\d+)/prospect/$',
    login_required(list_view.as_view(model=Prospect)),
    name = 'pre-pro-list'
))
urlpatterns = patterns(url(
    r'^(?P<company_id>\d+)/prospect/(?P<pk>\d+)/$',
    login_required(detail_view.as_view(model=Prospect)),
    name = 'pre-pro-detail'
))
urlpatterns = patterns(url(
    r'^(?P<company_id>\d+)/prospect/add/$',
    login_required(creation_view.as_view(model=Prospect)),
    name = 'pre-pro-add'
))
urlpatterns = patterns(url(
    r'^(?P<company_id>\d+)/prospect/(?P<pk>\d+)/edit/$',
    login_required(update_view.as_view(model=Prospect)),
    name = 'pre-pro-edit'
))
```

```
urlpatterns = patterns(url(
    r'^(?P<company_id>\d+)/prospect/(?P<pk>\d+)/delete/$',
    login_required(delete_view.as_view(model=Prospect)),
    name = 'pre-pro-delete'
))
```

### 2.2.3 Views

You can inherit from django-intranet `base_views` to customize your `change_list` display:

```
from intranet.base_views import IntranetListView

class ProspectListView(IntranetListView):
    list_display = ('company_name', 'last_name', 'category')
    list_display_links = ('company_name',)
    search_fields = (
        'company_name',
        'first_name',
        'last_name',
        'category__name',
    )
```

### 2.2.4 Templates

You will probably have to customize the `model_detail` template:

You can create `templates/<module_name>/<model>/model_detail.html`

For our app, we will create : `templates/prospect/prospect/model_detail.html` and `templates/prospect/category/model_detail.html`

```
{% extends 'intranet/model_detail.html' %}
{% load i18n intranet_extra intranet_values_extras %}
{% load url from future %}

{% block title_content %}{{ verbose_name|capfirst }} : {{ object|upper }}{% endblock %}
↪{%}

{% block content %}
<div class="row-fluid">
  <div class="span3">
    <table class="{% default_table_class %}">
      <tr><th colspan="2" style="text-align:center;">{% trans "General" %}</th></tr>
      <tr><th>{% trans "Company name" %}</th><td>{{ object.company_name|upper }}</td>
↪</tr>
      <tr><th>{% trans "Category" %}</th><td><a href="{% url 'category-detail' object.
↪category.pk %}">{{ object.category.name|title }}</a></td></tr>
    </table>
  </div>
  <div class="span3">
    <table class="{% default_table_class %}">
      <tr><th colspan="2" style="text-align:center;">{% trans "Contacts" %}</th></tr>
      <tr><th>{% trans "Company head" %}</th><td>
        {% if object.last_name == "" %}
          {{object.first_name|title }}
↪
```

```

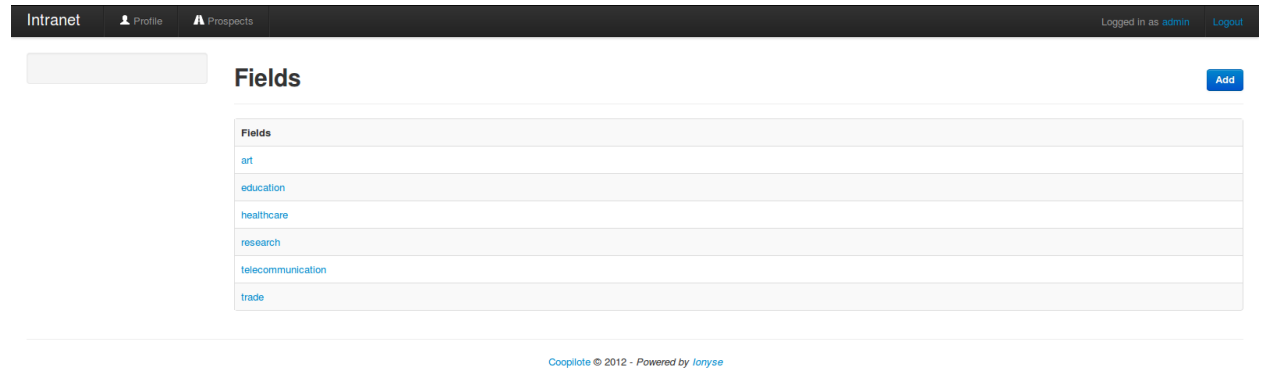
    {% else %}
        {{ object.title|title }} {{object.first_name|title }} {{ object.last_
↪name|upper }}
        {% endif %}
    </td></tr>
    <tr><th>{% trans "Phone" %}</th><td>{{ object.phone_number|phone }}</td></tr>
    <tr><th>{% trans "E-mail" %}</th><td><a href="mailto:{{ object.e_mail }}">{{_
↪object.e_mail }}</a></td></tr>
</table>
</div>
</div>
{% endblock %}

```

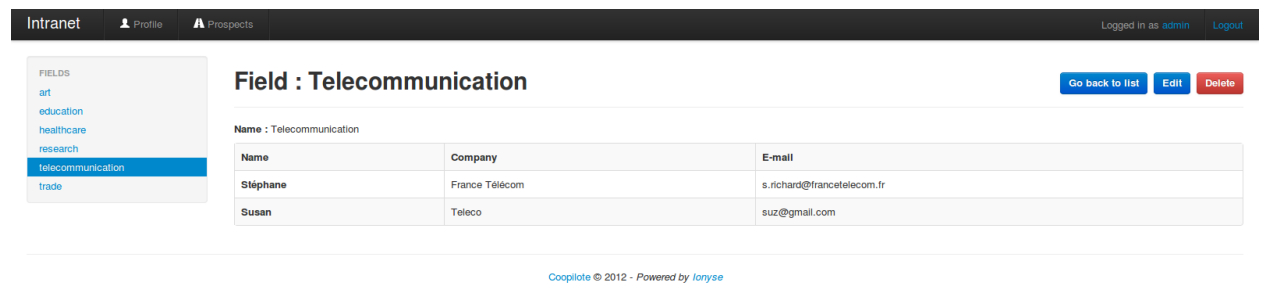
## 2.2.5 Screenshots

Here are some screenshots of the templates of a project using django-intranet. We defined two models: *Field* and *Prospect*.

Display of the *Field* list:



Display of a customized view of a *Field* entry:



Display of the *Prospect* list:

Intranet Profile Prospects Logged in as admin Logout

LOOKUP

**Prospects** Add

Company name	Last name	Field
France Télécom	Richard	telecommunication
Lentreprise	Wallet	trade
RGPST		education
Teleco	Lambert	telecommunication

Coopliote © 2012 - Powered by Ianyse

Display the details of a *Prospect* entry:

Intranet Profile Prospects Logged in as admin Logout

PROSPECTS  
France Télécom  
RGPST  
Teleco

**Prospect : FRANCE TÉLÉCOM** Go back to list Edit Delete

General		Contacts		Address	
Company name	FRANCE TÉLÉCOM	Company head	Mr. Stéphane RICHARD	Street	6 place D Alleray
SIRET code	380 129 866 00014	Profession	Chairman Of The Board Of Directors	Postal code	75015
Field	Telecommunication	Phone	08 99 23 34 89	City	Paris
		E-mail	s.richard@francetelecom.fr		

Coopliote © 2012 - Powered by Ianyse

Creation of a prospect entry:

Intranet Profile Prospects Logged in as admin Logout

**Creation of prospect**

Company name

Field

SIRET code

Title

First name

Last name

Profession

E-mail

Phone number

Please enter a phone number in the form: 03 90 87 65 43 or +33.390876543

Street

Postal code

City

Cancel Create

Coopliote © 2012 - Powered by Ianyse

Modification of a prospect entry:

Intranet Profile Prospects Logged in as admin Logout

### Edition of France Télécom

Company name

Field

SIRET code

Title

First name

Last name

Profession

E-mail

Phone number

Please enter a phone number in the form: 03 90 87 65 43 or +33.390876543

Street

Postal code

City

Coopilote © 2012 - Powered by Ionyse

## 2.3 Utils

### 2.3.1 Widgets

django-intranet provides you with extra widgets for your form fields:

- The PhoneField, extended from Django’s CharField, enforces the entered data of the field to be in the form of a valid telephone number, and then django-intranet will refuse to save a wrong phone number in your database. The telephone number can have a prefix code, for example “+81.0123456789”. If it doesn’t have one, django-intranet will consider we are in France by default. It will then remove separating characters like ‘ ’ or ‘.’, check the number of digits, and then save the number with the telephone code “+33.”.
- The SiretField, extended from Django’s CharField, enforces the entered data of the field to be in the form of a valid SIRET number. So django-intranet will check the number of digits and then use a simple algorithm to check if the entered number is a valid SIRET number.

### 2.3.2 Filters

You can also use the associated filters in your html templates:

- The filter “phone” transforms a number in the form “+81.0123456789” into “+810123456789” and a number in the form “+33.384289468” into “03 84 28 94 68” for better readability.
- The filter “siret” transforms a number in the form “01234567890128” into “012 345 678 90128” for better readability.